

## CLAIMS

What is claimed is:

1. A system for distributing a non-distributed computer code object having at least one public method in a distributed environment having middleware, the system comprising:
  - a) a server configured to extract at least one method corresponding to the object in the distributed computing environment;
  - b) the server further configured to generate distribution code to enable the distribution of the object; and,
  - c) the server further configured to generate integration code specific to the middleware onto which the object is integrated.
2. The system of claim 1, wherein the object is deployed as a publisher.
3. The system of claim 1, wherein the object is deployed as a subscriber.
4. The system of claim 3, wherein the server is configured to generate subscription specific code for the subscriber object, including method invocation code for subscribed methods of the publishers to which the subscriber object is subscribed, and the implementation of the Publisher's Java Interface.
5. The system of claim 1, wherein the server is configured to publish at least one method.
6. The system of claim 1, wherein the server is configured to allow the subscription of at least one method.
7. The system of claim 1, wherein the extracted method is made public by the object.

8. The system of claim 1, wherein the object is a Java program having a plurality of public methods, and the server is further configured to
  - a) extract the plurality of public methods of the object;
  - b) allow for the selection of at least one of the extracted public method; and
  - c) publish the selected extracted public method.
9. The system of claim 1, wherein the server is further configured to:
  - a) extract a plurality of public methods of at least one publisher object from a repository of middleware;
  - b) allow for the selection of at least one of the extracted public method; and
  - c) subscribe at least one subscribing object to the selected extracted public method.
10. The system of claim 9, wherein the configuration of the server to extract a plurality of public methods further comprises the server configured to:
  - a) locate and read the repository;
  - b) extract a list of publishers and published methods corresponding to the list of publishers from the repository;
  - c) prepare a storage object that stores the extracted contents of the repository; and
  - d) display the extracted contents of the repository from the storage object for selection by a user.
11. The system of claim 9, wherein the configuration of the server to allow for the selection of at least one of the extracted public method further comprises the server configured to:
  - a) display a plurality of public methods to a user;
  - b) allow the user to select at least one method from the displayed plurality of public methods through a user interface; and
  - c) prepare a storage object that stores the selected method.
12. The system of claim 9, wherein the repository is located remotely from the server.

13. The system of claim 1, wherein the object is in native binary code and the server is further configured to translate native binary code into Java bytecode.
14. The system of claim 13, wherein the configuration of the server to translate native binary code into Java bytecode further includes the server configured to generate a computer program containing an implementation of the object's published methods in C++ language and to generate a Java Native Interface implementation program in Java, containing the implementation of the object's published methods in C++ language.
15. The system of claim 1, wherein the object is a publisher object having published methods, and the configuration of the server to generate distribution code further comprises the server configured to:
  - a) generate a Java Interface containing the published methods; and
  - b) generate a Java implementation program, implementing the generated Java Interface, and containing a Remote Method Invocation Mechanism.
16. A method for dynamically distributing and integrating a non-distributed object in a distributed environment comprising the steps of:
  - a) selecting at least one method of the non-distributed object;
  - b) generating distribution code to enable the distribution of the object; and,
  - c) integrating the object in the distributed environment.
17. The method of claim 16, where the object is deployed as publisher object having public methods available for use by subscriber objects.
18. The method of claim 16, where the object is deployed as subscriber object.
19. The method of claim 16, wherein the object comprises a plurality of objects comprising subscriber objects and publisher objects.
20. The method of claim 19, further comprising:
  - a) extracting at least one method for each object;
  - b) storing methods selected for publishing and subscribing;

- c) for each non-Java object, generating translation code to translate the non Java object into a Java object;
  - d) generating distribution code for each publisher object;
  - e) generating subscription specific code for each subscribing object, wherein the subscription specific code contains subscribed methods of at least one publisher object;
  - f) generating integration specific code to integrate each object with middleware;
  - and
  - g) generating archive files for distributing a final output after compilation.
- 21. The method of claim 20, wherein at least one object is a Java program having a plurality of public methods, and the step of extracting at least one method for the object further comprises:
  - a) extracting a plurality of public methods of the object;
  - b) allowing for the selection of at least one of the extracted public method; and
  - c) publishing the selected extracted public method.
- 22. The method of claim 20, wherein at least one object is a subscriber object, and the step of extracting at least one method for the object further comprises :
  - a) extracting a plurality of public methods of at least one publisher object from a repository of middleware;
  - b) allowing for the selection of at least one of the extracted public method; and
  - c) subscribing the subscriber object to the selected extracted public method.
- 23. The method of claim 22, wherein the step of extracting a plurality of public methods of at least one publisher object from a repository of middleware further comprises
  - a) locating and reading the repository;
  - b) extracting a list of publishers and published methods corresponding to the list of publishers from the repository;
  - c) preparing a storage object that stores the extracted contents of the repository; and
  - d) displaying the extracted contents of the repository from the storage object for selection by a user.

24. The method of claim 20, wherein the step of extracting at least one method for each object further comprises the steps of:
  - a) displaying a plurality of public methods to a user;
  - b) allowing the user to select at least one method from the displayed plurality of public methods through a user interface; and
  - c) preparing a storage object that stores the selected method.
25. The method of claim 20, wherein at least one object comprises a native language publisher object, and the step for generating the distribution code further comprises translating native code into Java bytecode.
26. The method of claim 25, wherein the native language publisher object has published methods and the step of translating native code into Java bytecode comprises
  - a) generating a computer program implementing the published methods of the native language publisher object in C++ language; and
  - b) generating of a Java Native Interface implementation program in Java, containing the published methods of the native language publisher object in C++ language.
27. The method in Claim 26, where the implementation of the generated Java Native Interface comprises the steps of generating a Java implementation program containing the Remote Method Invocation.
28. The method in Claim 27 further comprising the steps of
  - a) generating implementation code implementing the Java Native Interface;
  - b) generating implementation code of the published methods; and
  - c) generating Remote Method Invocation code.
29. The method of claim 20, wherein the step of integrating the object further comprises the step of generating integration code specific to the distributed environment.
30. The method in Claim 20 wherein the step of storing methods selected for publishing and subscribing further comprises storing methods selected for publishing and subscribing in a local repository.

31. The method in Claim 20 wherein the step of storing methods selected for publishing and subscribing further comprises storing methods selected for publishing and subscribing in a remote repository.

10056708.012402